Appl. No. 10/631,195
Amdt. Dated July 6, 2006
Reply to Office Action of February 10, 2006

Amendments to the Specification

Please replace paragraph **[0001]** with the following amended paragraph:

[0001] This application claims priority to U.S. Provisional Application Serial No. 60/400,391 titled "JSM Protection," filed July 31, 2002, incorporated herein by reference. This application also claims priority to EPO Application No. 03291923.5, filed July 30, 2003 and entitled "Conditional Garbage Based On Monitoring To Improve Real Time Performance," incorporated herein by reference. This application also may contain subject matter that may relate to the following commonly assigned co-pending applications incorporated herein by reference: "System And Method To Automatically Stack And Unstack Java Local Variables," Serial No. [[__________]]10/632,228, filed July 31, 2003~~, Attorney Docket No. TI-35422 (1962-05401)~~; "Memory Management Of Local Variables," Serial No. [[__________]]10/632,067, filed July 31, 2003~~, Attorney Docket No. TI-35423 (1962-05402)~~; "Memory Management Of Local Variables Upon A Change Of Context," Serial No. [[__________]]10/632,076, filed July 31, 2003~~, Attorney Docket No. TI-35424 (1962-05403)~~; "A Processor With A Split Stack," Serial No. [[__________]]10/632,079, filed July 31, 2003~~, Attorney Docket No. TI-35425(1962-05404)~~; "Using IMPDEP2 For System Commands Related To Java Accelerator Hardware," Serial No. [[__________]]10/632,069, filed July 31, 2003~~, Attorney Docket No. TI-35426 (1962-05405)~~; "Test With Immediate And Skip Processor Instruction," Serial No. [[__________]]10/632,214, filed July 31, 2003~~, Attorney Docket No. TI-35427 (1962-05406)~~; "Test And Skip Processor Instruction Having At Least One Register Operand," Serial No. [[__________]]10/632,084, filed July 31, 2003~~, Attorney Docket No. TI-35248 (1962-05407)~~; "Synchronizing Stack Storage," Serial No. [[________]]10/631,422, filed July 31, 2003~~, Attorney Docket No. TI-35429 (1962-05408)~~; "Methods And Apparatuses For Managing Memory," Serial No. [[________]]10/631,252, filed July 31, 2003~~, Attorney Docket No. TI-35430 (1962-05409)~~; "Write Back Policy For Memory," Serial No. [[________]]10/631,185, filed July 31, 2003~~, Attorney Docket No. TI-35431 (1962-05410)~~; "Methods And Apparatuses For Managing Memory," Serial No. [[________]]10/631,205, filed July 31, 2003~~, Attorney Docket No. TI-35432 (1962-05411)~~; "Mixed Stack-Based RISC Processor," Serial No. [[________]]10/631,308, filed July 31, 2003~~, Attorney Docket No. TI-35433 (1962-05412)~~; "Processor That Accommodates Multiple Instruction Sets And Multiple Decode Modes," Serial No. [[________]]10/631,246, filed July 31, 2003~~, Attorney Docket No. TI-35434 (1962-05413)~~; "System To Dispatch Several Instructions On Available Hardware Resources," Serial No. [[________]]10/631,585, filed July 31, 2003~~, Attorney Docket No. TI-35444 (1962-05414)~~; "Micro-Sequence Execution In A Processor," Serial No. [[________]]10/632,216, filed July 31, 2003~~, Attorney Docket No. TI-35445 (1962-05415)~~; "Program Counter Adjustment Based On The Detection Of An Instruction Prefix," Serial No. [[________]]10/632,222, filed July 31, 2003~~, Attorney Docket No. TI-35452 (1962-05416)~~; "Reformat Logic To Translate Between A Virtual Address And A Compressed Physical Address," Serial No. [[________]]10/632,215, filed July 31, 2003~~, Attorney Docket No. TI-35460 (1962-05417)~~; "Synchronization Of Processor States," Serial No. [[________]]10/632,024, filed July 31, 2003~~, Attorney Docket No. TI-35461 (1962-05418)~~; "Inter-Processor Control," Serial No. [[________]]10/631,120, filed July 31, 2003~~, Attorney Docket No. TI-35486 (1962-05420)~~; "Cache Coherency In A Multi-Processor

System," Serial No. [[__________]]10/632,229, filed July 31, 2003~~, Attorney Docket No. TI-35637 (1962-05421)~~; "Concurrent Task Execution In A Multi-Processor, Single Operating System Environment," Serial No. [[__________]]10/632,077, filed July 31, 2003~~, Attorney Docket No. TI-35638 (1962-05422)~~; and "A Multi-Processor Computing System Having A Java Stack Machine And A RISC-Based Processor," Serial No. [[__________]]10/631,939, filed July 31, 2003~~, Attorney Docket No. TI-35710 (1962-05423~~).

Please replace paragraph **[0021]** with the following amended paragraph:

[0021] Figure 2 shows an exemplary block diagram of the JSM 102. As shown, the JSM includes a core 120 coupled to data storage 122 and instruction storage 130. The core may include one or more components as shown. Such components preferably include a plurality of registers 140, three address generation units ("AGUs") 142, 147, micro-translation lookaside buffers (micro-TLBs) 144, 156, a multi-entry micro-stack 146, an arithmetic logic unit ("ALU") 148, a multiplier 150, decode logic 152, and instruction fetch logic 154. In general, operands may be retrieved from data storage 122 or from the micro-stack 146, processed by the ALU 148, while instructions may be fetched from instruction storage 130 by fetch logic 154, pre-decoded by pre-decode logic ~~158~~, and decoded by decode logic 152. The address generation unit 142 may be used to calculate addresses based, at least in part on data contained in the registers 140. The AGUs 142 may calculate addresses for C-ISA instructions as will be described below. The AGUs 142 may support parallel data accesses for C-ISA instructions that perform array or other types of processing. AGU 147 couples to the micro-stack 146 and may manage overflow and underflow conditions in the micro-stack preferably in parallel. The micro-TLBs 144, 156 generally perform the function of a cache for the address translation and memory protection information bits that are preferably under the control of the operating system running on the MPU 104. The decode logic 152 may be adapted to execute both the standard Java instructions as well as the C-ISA instructions of the system.

Please replace paragraph **[0032]** with the following amended paragraph:

[0032] In either previously described embodiments, the value of the memory usage counter ~~160~~ 168 may be compared to that of a threshold value 166, and upon approaching the threshold, the JSM 102 may send to the MPU 104 an interrupt signal, which indicates the need for garbage collection. The MPU 104 may then initiate the garbage collector 158 in the memory 106, where the garbage collector may evaluate memory 106. After freeing consumed memory resources that may not be needed, the memory usage memory usage counter 168 may be updated. The memory usage counter 168 may now again be used to monitor the memory consumption and may be incremented as described above.